



04

## El modelo, el controller y el DAO de categorías

### Transcripción

[00:00] Hola, ya atendimos el primer requisito de nuestro cliente que es relacionar cada producto a una categoría. Lo hicimos desde la base de datos, pero ahora necesitamos hacerlo en la aplicación también. Como creamos una nueva tabla en la base, nosotros necesitamos desarrollar las capas de controlar y model para esta entidad en la aplicación, recordando que el controller hace la conexión entre la view y la fuente de datos que contiene el modelo.

[00:27] Vamos a nuestra aplicación y completar las funcionalidades del listar categorías en la caja de selección y poder utilizar la categoría elegida para poder guardar un nuevo producto. Ahora, en la opción acá de categoría solamente tenemos una opción que dice: Elige una categoría. Vamos acá al código para ver cómo funciona la población de este campo en el formulario.

[00:51] Acá en Eclipse en el ControlDeStockFrame, nosotros acá arriba tenemos un JComboBox

[01:18] Entonces aquí en el paquete modelo nosotros vamos a hacer un new class y vamos a crear una clase de categoría también. Categoría, ahí está, la creamos. Aquí en ella vamos a agregar los mismos campos que tenemos en la tabla, que si vemos aquí en la consola son id y nombre, entonces voy viendo el Eclipse, vamos a declarar el private Integer id; private String nombre.

[01:46] Era nombre, ¿no? A ver nombre, nombre de la categoría. Por ahora es eso, no vamos a crear los getters y setters porque lo que estamos haciendo aquí

es ir creando estos métodos de acorde a nuestra necesidad para no estar creando montón de código que nos vamos a estar utilizando.

[02:05] Ahora que creamos la clase categoría, podemos volver aquí a ControlDeStockFrame y ya cambiar este object para categoría. Ahora sí, y la importamos. Listo. Ya realizamos el primer cambio. Ahora Eclipse empieza a indicar algunos puntos de error en el código. Entonces vamos a hacer un clic aquí para ver dónde en dónde es el error que queda aquí en el método configurar campos del formulario.

[02:33] Aquí en este método lo que estamos haciendo es un comboCategoría que agrega un ítem que es un string que dice: "Elige una categoría" y también hay una parte de código acá que está con el comentario de TODO que es el por hacer. Entonces aquí tenemos que hacer algunas mejoras ya también.

[02:53] Lo primero aquí que vamos a hacer es, en este addItem vamos a hacer en lugar de estar enviando un string de la categoría, ya que el comboCategoría ahora no es más de objeto, vamos a agregar un new Categoria. Y aquí ponemos un id 0 que no existe, y la string de categoría.

[03:16] Como este constructor no existe, nosotros vamos a crearlo en la clase. Vamos a crear acá el constructor y luego vamos a asignar, voy a mover este método de acá para abajo, para dejar más ordenado. Y ahora voy a primero cambiar los nombres de los parámetro y asignarlos a los atributos de la clase.

[03:38] Entonces this.id = id; this.nombre = nombre. Y listo. Inicializamos ya la categoría para poner el primer elemento de nuestro listado de categorías en la cajita de selección. Ahora en ese pedazo acá de TODO, lo que vamos a hacer es completar la lógica para buscar las informaciones, acá sí entramos en listar, nosotros vamos a completar la lógica para buscar las informaciones de categoría directamente en la base de datos.

[04:09] Bueno, estamos aquí en categoría controller, y lo primero que vamos a hacer ya es cambiar esto de un listado de cualquier cosa que está devolviendo

para hacer un listado de categoría. Entonces ya sabemos que el categoría controller, el método listar devuelve un listado de categoría.

[04:27] Y ahora, tal como hicimos con el modelo de producto, nosotros vamos a crear la clase de categoría DAO para abstraer toda la lógica de conexión con la base de datos referente al modelo de categoría. Entonces aquí ya voy a crear un `private CategoríaDAO categoríaDAO;` y por ahora, va a quejarse un poco Eclipse, pero deja que se queje un poco y ya, ya vamos creando los objetos.

[04:56] Entonces aquí creamos la categoríaDAO como un atributo de la clase. Y como hicimos en el productoController, vamos a crear un constructor acá de CategoríaController y en él nosotros vamos a inicializar la categoríaDAO, entonces vamos a hacer aquí un `var de factory` para tomar nuestro `ConnectionFactory();` a ver, ahora sí, `ConnectionFactory();`

[05:24] Y vamos a decir que `this.CategoríaDAO = new CategoríaDAO,` que recibe nuestra `(factory.recuperaConexion());` entonces estamos inicializando el DAO enviando una conexión del pool de conexiones. Ahora sí vamos a ¿hacer qué? ¿Crear el objeto? Aún no. Vamos ya a adelantar un poco más y en lugar de un `return ArrayList` vamos a hacer aquí `categoríaDAO.listar();`

[06:01] Aquí ya tenemos toda la funcionalidad del controller, solamente nos falta crear el DAO. Entonces vamos a hacerlo, hacemos aquí un control o comando 1 para crear la clase de categoría DAO. Aquí vamos a cambiar el paquete para poner en lugar de controller, el DAO. Hacemos un finish. Ya tenemos la categoría.

[06:24] Ahora la clase, perdón, ahora voy viendo aquí a nuestro controller, vamos a crear el constructor. Ahí lo tenemos. Y por último, vamos a crear el método listar. Ahora sí, ya tenemos todo, acá tenemos que inicializar nuestra categoríaDAO para poner la conexión a un atributo de conexión y poder desarrollar en el método del listado.

[06:50] Entonces vamos acá, declarar un private Connection del paquete java.sql, con; y acá vamos a cambiar el nombre y this.con = con; ya tenemos inicializado. Acá en el método de listar ya vamos a hacer lo siguiente, Vamos a empezar de a poquito, vamos a hacer crear primero nuestro resultado, que es un List<categoría>

[07:22] Vamos a decir que es el resultado = new ArrayList<>(); ahí lo tenemos. Y devolvemos este resultado. Por ahora es eso, ya vamos a poner la lógica para acceder a la base de datos. Y ahora lo que vamos a hacer acá es como hicimos en los otros metodos del producto DAO, crear un PreparedStatement.

[07:48] Ahí lo importamos statement = con.prepareStatement. Y acá en el comando SQL nosotros vamos a hacer nuestro select de los datos de categoría, entonces es "SELECT ID, NOMBRE FROM CATEGORÍA"). Este es nuestro SELECT, ya está el comando.

[08:12] Ahora lo que tenemos que hacer es, ya se queja Eclipse, que es el try catch. Ya tenemos acá el SQLException, hacemos un throw new RuntimeException(e); con la SQLException ahí encapsulada, y listo. Okay, ahora vamos a hacer algo distinto.

[08:32] Acá, en lugar de estar haciendo como siempre hicimos de statement.execute, que devuelve un boolean y después tenemos que hacer un statement.getResultSet para tomar los resultados, vamos a utilizar un acceso directo para ejecutar la query y ya tomar el resultado.

[08:50] Este método que tiene ese acceso directo se llama executeQuery. Así de simple, executeQuery ya nos devuelve a ver, acá está el try-with-resources, acá el resultSet. Pero como yo estoy siguiendo la sintaxis del Java 9, voy a hacer un final acá de resultSet, ahora sí, y el executeQuery nos devuelve ese resultado.

[09:24] Y acá yo pongo en lugar de executeQuery acá, el resultSet. Ahora sí, con resultSet. Ahí está. Bueno, así como hicimos el resultSet adentro de un try, nosotros también tenemos que hacer el try-with-resources de statement.

Entonces acá los declaro como final y ahora sí el statement ya también está dentro de lo que necesitamos.

[09:58] Y ahora con este resultSet, nosotros vamos a tomar cada fila de resultado para crear un objeto del tipo categoría y agregar al listado de resultados. ¿Cómo lo hacemos? Hacemos un while (resultSet.next) ¿Se acuerdan de eso? Hace mucho. Bueno, (resultSet.next) y ahí tomamos el new Categoría(resultSet.getInt de la columna "ID" y de resultSet.getString de la columna "NOMBRE").

[10:42] Ahí está, vamos a asignar eso a una nueva variable, var categoría = new Categoría. Ya la tenemos acá y, bueno, último, la agregamos a nuestro resultado.add(categoría); listo, perfecto, listo. Guardamos estos cambios de acá y volvemos para la clase ControlDeStockFrame.

[11:07] Ya tenemos acá el resultado de categorías y tenemos un pedazo de código acá comentado que tiene un categorías.forEach en donde estamos agregando todos los ítems de categoría. Acá saqué el comentario y listo, vamos a hacer las pruebas ahora, voy a guardar todo y vamos a hacer estas pruebas. Okay.

[11:28] Aquí abrimos la aplicación, pero en la caja de categorías estamos imprimiendo acá unos hashes del objeto, de la categoría. Dejamos pasar algún detalle acá, porque recuerda que antes estábamos agregando solamente el texto y funcionaba. Puede ser que esté pasando algo acá, vamos a hacer lo siguiente.

[11:51] Acá voy a cerrar la aplicación y en categoría vamos a sobrescribir el método toString. Voy a escribir acá toString y lo voy a sobrescribir y en lugar de llamar el super voy a devolver this.nombre. Ahora sí, vamos a probar una vez más. Bueno, ahora sí está mostrando la categoría, los muebles acá, tecnología, cocina, zapatillas. Muy bien. Ahora vamos a hacer lo siguiente.

[12:21] Este mouse inalámbrico voy a eliminarlo y voy a registrarlo una vez más porque estaba todo mal escrito, entonces vamos a poner acá mouse inalámbrico, 10 cantidades y la categoría tecnología. Voy a guardar y ahí está insertado con éxito. Vamos a la consola para ver si realmente insertó con el id correcto, vamos a ver acá.

[12:53] Voy a limpiar la consola, hacemos un `SELECT * FROM PRODUCTO` y tenemos el mouse inalámbrico con la categoría nula. ¿Qué pasó? Lo que pasó es que aún nos falta desarrollar la lógica para vincular el id de la categoría al producto. Por ahora ya hicimos un montón.

[13:18] Creamos acá la capa de model en el proyecto, tenemos la categoríaDAO, la categoría también, ya tenemos la capa de model, tenemos también la capa de controller y ya estamos agregando también el listado de las categorías en nuestra view acá, en nuestra vista.

[13:36] Ya tenemos concluida una tarea más del requerimiento de nuestro cliente. En la próxima clase vamos a desarrollar el próximo requerimiento que es el de asignar esta categoría que elegimos en la pantalla al producto que estamos registrando también.